Video Background Music Generation: Dataset, Method and Evaluation

Le Zhuo[1*] Zhaokai Wang[1*] Baisen Wang[1*] Yue Liao[1†] Chenxi Bao[1,2]
Stanley Peng[1] Songhao Han[1] Aixi Zhang[3] Fei Fang[3] Si Liu[1]
[1]Beihang University [2]Edinburgh College of Art, University of Edinburgh [3]Alibaba Group
zhuole1025@gmail.com, {wzk1015, wbs2788}@buaa.edu.cn, liaoyue.ai@gmail.com

Abstract

Music is essential when editing videos, but selecting music manually is difficult and time-consuming. Thus, we seek to automatically generate background music tracks given video input. This is a challenging task since it requires music-video datasets, efficient architectures for video-to-music generation, and reasonable metrics, none of which currently exist. To close this gap, we introduce a complete recipe including dataset, benchmark model, and evaluation metric for video background music generation. We present SymMV, a video and symbolic music dataset with various musical annotations. To the best of our knowledge, it is the first video-music dataset with rich musical annotations. We also propose a benchmark video background music generation framework named V-MusProd, which utilizes music priors of chords, melody, and accompaniment along with video-music relations of semantic, color, and motion features. To address the lack of objective metrics for video-music correspondence, we design a retrieval-based metric VMCP built upon a powerful video-music representation learning model. Experiments show that with our dataset, V-MusProd outperforms the state-of-the-art method in both music quality and correspondence with videos. We believe our dataset, benchmark model, and evaluation metric will boost the development of video background music generation. Our dataset and code are available at https://github.com/zhuole1025/SymMV.

1. Introduction

Music plays a crucial role when creating videos, improving the overall quality of videos and enhancing the immersion for viewers. With the rapid growth of social platforms, the needs to find suitable music for videos extend from professional fields, *e.g.*, soundtrack production in the film industry, to amateur usages like video blogs and TikTok short videos. However, finding proper music for videos and making alignments are difficult and may even bring copyright issues. Thus, automatically generating background music for videos is of great value to a wide range of creators.

Recently, tremendous progress has been made with text-to-image systems [37, 39], revealing the powerful generative capacity of state-of-the-art models. Though available to the same family of generative models, the area of video-to-music generation is still in the preliminary stage. We attribute this to the following three key challenges from the aspects of dataset, method, and evaluation, respectively: (1) Large-scale datasets of high-quality music with paired videos are absent and cumbersome to collect. (2) Video-music correspondence is complex and tricky to integrate effectively into current generative models. (3) There is a lack of objective metrics to evaluate the correspondence between video and music. We give detailed explanations and provide the corresponding solutions for each challenge in the rest of our paper. Our framework is illustrated in Fig. 1.

Existing video-music datasets [20, 32, 56] are either limited in size or weak in video-music correspondence due to noisy data pairs. More importantly, these datasets only include music in audio format, which is complex, computationally expensive, and difficult to impose control signals from videos. On the contrary, symbolic music, representing music in discrete sequence [24, 21], contains rich semantic information and helps to explore video-music relationships.

To fill this gap, we introduce a novel video and symbolic music dataset named Symbolic Music Videos (SymMV). It contains piano covers of popular music with their official music videos carefully collected from the Internet. Overall, it contains 1140 video-music pairs of more than 10 genres with a total length of 76.5 hours. As an advantage of symbolic music, we provide chord, melody, accompaniment, and other metadata for each music. The detailed annotations allow model to decouple the music generation process into stages and better control the generated music. Note that music in SymMV is of high quality and can also be directly used for unconditional music generation without video modality.

We then explore the method for video background music

*Equal contribution.
†Corresponding author.

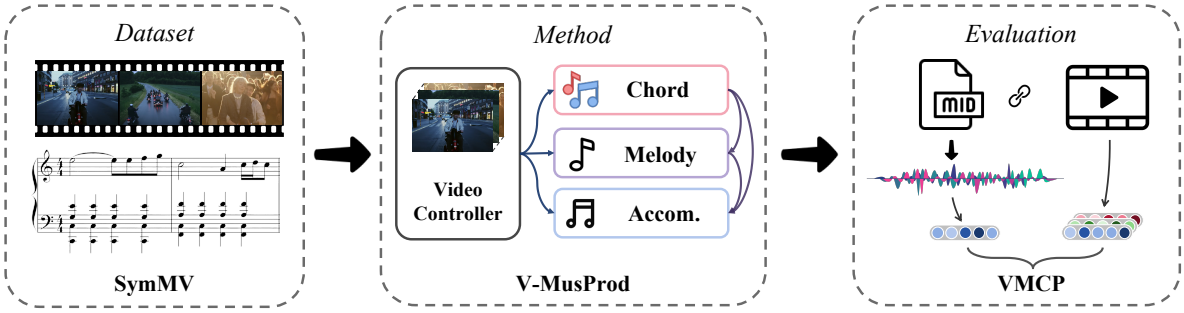


Figure 1: **Overview of our framework.** We solve the task of video background music generation from three perspectives. **Left:** We present the first video and symbolic music dataset with detailed annotations. **Middle:** We decouple music generation into three progressive stages: chord, melody, and accompaniment (accom.), and extract various video features to guide different generation stages. **Right**: We propose a novel evaluation metric, named Video-Music CLIP Precision (VMCP), to measure the correspondence between generated music and input video.

generation. It is challenging since the correspondence between music and video is not a deterministic one-to-one mapping but a more complex one related to aesthetic style. Models are required to create music that is not only coherent and melodious but also harmonic with the given video in terms of both rhythm and style. Some initial attempts [45, 56] solve the motion-to-music generation via rhythmic control by motion features in videos. They suffer from serious limitations on applied video types, *i.e.*, requiring additional key points annotation. Prior work CMT [9] designs rule-based rhythmic relationships to generate video background music due to lack of paired data. It ignores the semantic-level correspondence, which sometimes results in conflicting styles.

To build a benchmark model, we propose a Video Music generation framework with Progressive decoupling control (V-MusProd). V-MusProd decouples music generation into three progressive transformer stages: chord, melody, and accompaniment. It first predicts a chord sequence, then generates melody conditioned on chords and finally generates accompaniment conditioned on chords and melody. We extract semantic and color features to control the Chord Transformer since these features reflect the theme and emotion of a video. Motion features are extracted as rhythmic control for Melody and Accompaniment Transformers since their outputs require rhythmic alignment with the input video. V-MusProd can generate melodious music corresponding to the input video based on the two controlling processes.

Lastly, another bottleneck of video background music generation lies in the inadequate objective metrics. Previous methods [9, 56, 45] use metrics for unconditional music generation or subjective evaluation to examine the video-music correspondence. Common metrics for unconditional music generation can only assess the quality and diversity of the generated samples but ignores the alignment between paired music and videos. Besides, conducting human listening tests for subjective evaluation is cumbersome and sometimes biased. Recent text-to-image generation models [34, 39] leverage the powerful pretrained vision-language CLIP [36] model to compute the similarity between input prompts and generated images, opening up new opportunities for evaluating sample correspondence. Therefore, we propose a new evaluation metric, named Video-Music CLIP Precision (VMCP), which extends the vision-language CLIP model to video and music domain to measure the video-music correspondence. With VMCP and subjective evaluation, V-MusProd demonstrates better results than CMT on SymMV.

Our contributions are summarized as follows: (1) We present SymMV, the first video and symbolic music dataset with detailed annotations tailored for video background music generation. (2) We propose V-MusProd, a benchmark framework that utilizes music priors of chords, melody, and accompaniment along with video-music relations of semantic, color, and motion features. (3) We propose an objective metric VMCP based on the video-music CLIP model. Both objective metrics and subjective evaluation demonstrate that V-MusProd surpasses the state-of-the-art model in video-music correspondence and music quality.

2. Related Work

Video-music Datasets. Multi-modal datasets, *e.g.*, image-text [41, 6, 42], video-text [33, 55], video-audio [7, 15, 30, 32] greatly push the development of multi-modal learning tasks. However, there lacks datasets for video-conditional music generation. HIMV [20] is a large-scale dataset with 200K video-music pairs from YouTube-8M [1]. Despite its size, it suffers from weak correspondence and poor quality of music and videos, *e.g.*, static videos, live performances, or amateur montages, since it is only designed for the task of retrieval. Recently, AIST++ [32] and TikTok dataset [56] are introduced for music-dance learning, which contains videos, motion sequences, and paired music. However, their music parts are limited in size, *i.e.*, less than 5 hours, and are only

Dataset	Video	Audio	MIDI	Genre	Chord	Melody	Tonality	Video Content	Size	Length (Hours)
MAESTRO [17]	✗	✓	✓	✗	✗	✗	✗	-	1,276	198.7
POP909[50]	✗	✗	✓	✓	✓	✓	✓	-	909	70.0
HIMV-200K[20]	✓	✓	✗	✗	✗	✗	✗	Music Video	200,500	-
TikTok[56]	✓	✓	✗	✗	✗	✗	✗	Dance Video	445	1.5
AIST++[32]	✓	✓	✗	✓	✗	✗	✗	Dance Video	1,408	5.2
URMP[31]	✓	✓	✓	✗	✗	✗	✗	Music Performance	44	33.5
SymMV (Ours)	✓	✓	✓	✓	✓	✓	✓	Music Video	1,140	76.5

Table 1: **Comparison between different music datasets.** The proposed SymMV is the first dataset includes video and symbolic music pairs for video background music generation. Our dataset also provides various musical annotations and metadata such as genre, chord, and melody. We include popular symbolic music datasets in the first two rows for reference.

provided in audio format. Therefore, it is important to build a video-music dataset tailored for music generation. Referring to symbolic music generation literature[50, 22, 21], we build SymMV, which includes 1140 piano music in MIDI format with paired music videos and rich musical annotations.

Video-Conditional Music Generation. Most music generation methods fall into the unconditional setting [8, 22, 24, 21]. Previous video-conditional music generation works mainly focus on reconstructing music from silent instrument performance videos [44, 14, 43, 27]. Recent methods [45, 56, 57] are proposed to generate music for dancing or human activity videos based on rhythmic relations. However, their methods require extra motion annotations as inputs and thus do not work for general videos with various contents. CMT [9] generates background music for general videos with video rhythmic features, but its video-music relationships are purely rule-based, and it lacks semantic relationships between videos and music.

Objective Music Metrics. Previous objective music metrics [12, 11, 25] mainly focus on statistics of music features, where the performance of generated music is decided by their closeness to the training data. Text-to-image generative models use R-precision [52] to evaluate whether the generated image can be used to retrieve its input text description. Recent works [35, 10] improve the robustness of R-precision using the powerful CLIP model [36]. Cosine similarity computed by CLIP, named CLIPScore [18], can also be directly used as an objective metric.

3. Dataset

We collect the first video-music dataset that matches music videos and their piano version music in symbolic form. The collected SymMV dataset contains 1140 pop piano music in both MIDI and audio format with the corresponding official music video with a total duration of 76.5 hours. We split SymMV into the training set (1000 pairs), validation set (70 pairs), and test set (70 pairs). Our dataset also includes various annotated metadata, such as chord progression, tonality, and rhythm. Fig. 2 shows an example of our dataset.

Tab. 1 provides comparisons with existing video-music datasets.

3.1. Data Collection

Mass amounts of video-music pairs are available on the Internet. In particular, music videos have strong video-music correspondence in artistic styles and rhythms. They also have plenty of scenes, movements, and camera angles, thus suitable for learning intrinsic video-music relations. Therefore, we aim to construct a music video and paired symbolic music dataset for video background music generation.

Although finding music videos or their piano covers separately can be easy, it is hard to collect corresponding pairs. We solve this challenge by first collecting piano covers with fair to high audio and musical quality from professional piano tutorial YouTube channels. After downloading both audio and their metadata, we parse the metadata and use the parsed song title and singer as keywords to search for the corresponding official music video. Once video and audio pairs are collected, we transcribe the audio files into MIDI format using a state-of-the-art automatic piano transcription model [16]. To ensure dataset quality, we invite three professional musicians to manually check the collected dataset. They filter out low-quality pairs, such as static or lyric videos and fragmented or undesirable MIDIs, and find the resulting accuracy in note pitch and duration satisfactory.

3.2. Data Annotations

Melody and Accompaniment. Common pop music is well structured and can be decoupled into melody and accompaniment. *Melody*, a combination of pitch and rhythm, constitutes the most memorable aspect of a song. It is easier for people to perceive melody than other music parts. Hence, melody plays an essential role in a music piece. *Accompaniment* is correlated with the chord sequence and melody, serving as the background sound effect to harmonize the foreground melody [28], and can bring different auditory

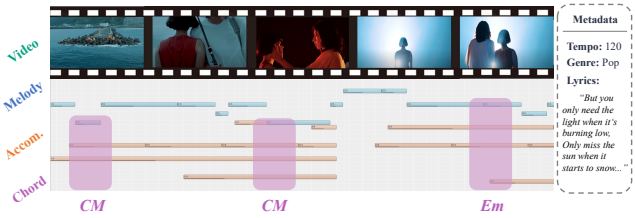


Figure 2: **Illustration of sample in SymMV.** Sample in our SymMV dataset includes paired music and video, music feature annotations, and related metadata.

sensations. Given music note sequences, we first quantize their duration to the 16th note and implement the Skyline algorithm [47] to separate the melody and accompaniment.

Chord Progression. *Chords*, several notes in a certain vertical configuration that sounds harmonic, run through the whole music piece and play an important role in setting the base tone of music. Moreover, chords convey strong emotions with several unique features like color and tension, *e.g.*, major chords bring a feeling of brightness, while minor chords sound relatively dim [40]. We provide chord progression to further control the music generation process. We adopt an open-source rule-based algorithm [1] to extract chords from MIDI files. We observe a long-tail distribution of chord progressions, where more than half of the chords occur less than 10 times. To mitigate this problem, we narrow down the chord templates to 12 root notes and 10 qualities, which covers mostly used types in pop songs.

Tonality. *Tonality* is the general term for *tonic* and *mode* of a key. It reflects the hierarchy of stability, attractions, and directionality in music work. The tonic chord is considered to be the most stable chord in tonality, and it determines the name of the key. Mode represents a type of musical scale centered on the tonic, which can be mainly divided into two types: major and minor modes. In our dataset, we provide the tonality annotation using Krumhansl-Schmuckler algorithm [29] to predict tonality from MIDI files and represent music keys using 12 tonic and 2 mode types.

Rhythm. We estimate the beat and downbeat positions from audio using the RNN-based model [5], which corresponds to the fine-grained rhythm. Then, we calculate the tempo from beat and downbeat positions to represent the global rhythm.

Metadata. We also provide additional metadata of our music dataset, such as genre and lyrics. We use ShazamIO[2] to search for lyrics and genres of music in SymMV. These metadata are helpful for data analysis and may benefit future applications, *e.g.*, text-to-music generation [3, 23].

[1]https://github.com/joshuachang2311/chorder/
[2]https://github.com/dotX12/ShazamIO

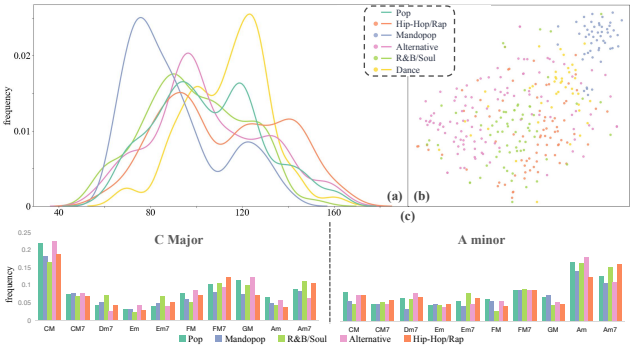


Figure 3: **Visualization of data statistics.** (a) Probability density function of beats per minute in different genres. (b) T-SNE visualization of visual features and genres. (c) Chord distribution in different genres. We show the high correlation between not only music and genre but also video and genre.

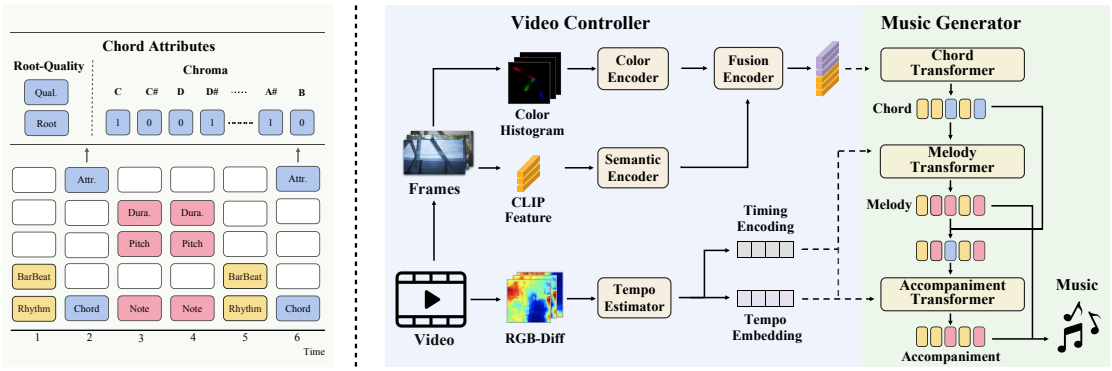
3.3. Data Analysis

To ensure the quality of our dataset and determine video-music relationships , we provide a detailed analysis of different music and video features. Genre, an attribute shared by both modalities, is convenient for us to analyze with visualization tools. Since our dataset contains video-music pairs with more than 10 genres, we use genre as a bridge between video and music to explore their distinct features. We transpose all major music to C major (C) and minor music to A minor (a) to remove the influence of tonality on our analysis.

Chord and Genre. We count the frequency of chords in different genres of music. To ensure statistical significance, we choose the five most frequent genres and ten chord types with high frequency and high variance. As Fig. 3 (c) shows, the final chord distribution meets our expectations. In *pop* music, the most steady chords, *e.g.*, CM, FM, and Am, occupy the largest proportion, in line with the stability of pop music. In contrast, some rarely seen chords have a high frequency in *R&B/Soul*, such as Dm7 and Am7, in order to create a richer and more diverse harmonic palette. *Alternative*, as a type of *Rock*, tends to favor simpler chords, so the occurrences of seventh chords are less frequent. As for *Hip-Hop/Rap*, the singing part is less melodic, and therefore, there are generally more seventh chords to provide accompaniments with more space for expression and to fill the harmony space. We provide more analysis in the Appendix.

Rhythm and Genre. We use kernel density estimate (KDE) to visualize the distribution of beats per minute (BPM) of music in various genres. As shown in Fig. 3 (a), *Dance* and *Hip-Hop/Rap* tend to have higher BPM. Curves in other genres display two distinct peaks, representing the BPM of slow-paced and fast-paced songs, respectively. Notably, there is no obvious peak corresponding to fast songs in *R&B/Soul*.

Visual Features and Genre. As for visual features, we first



(a) Music Representation

(b) Pipeline of V-MusProd

Figure 4: **Illustration of our method V-MusProd. Left**: We group multiple music attributes into one event-based token (each column). Different colors indicate different types of tokens. **Right**: We extract semantic, color, and motion features to guide the music generation process. The three types of features serve as inputs for different stages of the decoupled music generation model, which contains Chord, Melody, and Accompaniment Transformers.

extract 512-dimensional CLIP features from video frames at an FPS of 6. Then we compute the average of these features at time dimension to generate the visual feature of the whole video. We use t-SNE [48] to project visual features into a 2-dimensional space. We ignore *Pop* due to its complexity and select CLIP features of the other five genres to conduct the cluster analysis. Visual features are generally clustered by genre in Fig. 3 (b), demonstrating high correlations.

4. Method

We propose a novel music generation framework named V-MusProd, to tackle the challenging video background music generation task. Our framework is shown in Fig. 4, which consists of a video controller and a music generator. The video controller extracts visual and rhythmic features and fuses them as the contextual input of the music generator. The music generator decouples the music generation process into three progressive stages that are independently trained: Chord, Melody, and Accompaniment. At inference time, melody and accompaniment tracks are merged together to form a complete music piece. The progressive generation pipeline allows for the use of decoupling control on different generation stages, which improves the correspondence between videos and music. We elaborate on each component in the following subsections.

4.1. Video Controller

Directly using raw video frames as conditional input is difficult for model to learn the correspondence between two different modalities. Thus, it is important to design and extract meaningful features from video as intermediate representations to simplify the learning process. Considering the style and rhythm relationship between music and video, we extract semantic, color, and motion features separately to guide the music generation model.

Semantic Features. We use pretrained CLIP2Video [13] as the extractor to encode raw video frames into semantic feature tokens without finetuning. The CLIP2Video model builds upon the CLIP encoder [36], which is pretrained on billions of image-text pairs, and further uses a temporal difference block to learn the temporal context across frames. The extracted features are supposed to contain representations of different video semantics, *e.g.*, scenery, sports, and crowds, which are closely related to the content of music.

Color Features. Color in videos can reflect the underlying emotions in a given scene, corresponding with the mood of paired music. We employ color features as one of the control signals for chord generation. Specifically, we extract the color histogram of each video frame, *i.e.*, a 2D feature map proposed in [2], to represent the color distribution in a non-linear manifold. The color histogram projects an image's color into a log-chroma space, which is more robust and invariant to illumination changes.

Semantic and color features are fed into separate transformer encoders and then concatenated together at the length dimension. We add a learnable embedding to mark whether each token is from color feature or semantic feature, and feed the sequence into a transformer encoder for inter-modality and temporal fusion. The fused output serves as keys and values of cross attention in Chord Transformer.

Motion Features. We compute RGB difference as motion features to determine the music tempo and calculate Tempo Embedding and Timing Encoding. We extract the RGB difference with intervals of 5 frames (0.2 seconds) and map the mean RGB difference of a video to the music tempo. We use a linear projection from the minimum and maximum RGB difference to the tempo range of $[90, 130]$. The esti-

mated tempo is used as the Tempo Embedding in the music generator. We also add Timing Encoding [9] in Melody and Accompaniment Transformers to synchronize the video timing and music timing, which reminds the model of the current token's position in the whole sequence.

4.2. Music Generator

The music generator G, consisting of a Chord Transformer G_c, a Melody Transformer G_m, and an Accompaniment Transformer G_a, is designed to generate symbolic music conditioned on the extracted video feature. The workflow can be written as follows:

$$\begin{aligned} x_m &= G_m(G_c(y_s), y_r), \\ x_a &= G_a(G_c(y_s), x_m, y_r), \\ x &= x_m \oplus x_a, \end{aligned}$$

where style feature y_s and rhythm feature y_r is produced by video controller C, the final music piece x is composed of the melody x_m and the accompaniment x_a, $\oplus$ represents merging the two parts into a single track.

Music Representation. Symbolic music comprises a set of music attributes. To encode the dependencies among different attributes, we design an event-based music representation inspired by [21, 38]. We define three types of tokens, namely Note, Rhythm, and Chord, as the red, yellow, and blue columns in Fig. 4 (a), respectively. Each token is a stack of attributes. In particular, the Rhythm token comprises the BarBeat attribute, indicating the beginning of each bar or beat; Note token contains the Pitch and Duration attributes; Chord token contains the Root and Quality attributes, *i.e.*, the root note and the quality of chords. Chord can also be represented as chromagrams, a 12D binary vector where each dimension indicates whether a pitch class is activated. An additional Type attribute is applied for all tokens to mark their types. In our implementation, the Chord Transformer only models the Rhythm and the Chord tokens, while the Melody and Accompaniment Transformers model all three token types. To align the generated music with input video with rhythmic information, we add Bar Embedding and Beat Embedding for the absolute bar and beat position of current token, and Tempo Embedding for the music tempo.

Chord Transformer. We adopt a transformer decoder architecture for Chord Transformer to learn the long-term dependency of input video feature sequences. The event-based token sequence is added with positional encoding and fed into the Chord Transformer as a query. Meanwhile, the style features from video controller are fed as keys and values.

Melody Transformer. We employ an encoder-decoder transformer architecture for melody generation. The encoder receives a chord sequence as input, and then the decoder generates a note sequence as the output melody. Considering the relatively short-range dependency between melody and chords, we adopt a bar-level cross-attention mask so that each decoder token can only attend to the contextual encoder output within the previous current or next bar.

Accompaniment Transformer Similarly, we also adopt an encoder-decoder transformer to generate the accompaniment sequence. Since accompaniment closely correlates with chords and melody, we merge the generated chord sequence with the melody and then pass the merged sequence to Accompaniment Transformer as conditional input. We also apply the same bar-level cross-attention mask as in Melody Transformer. Eventually, the generated accompaniment is directly merged with the melody to form the final music.

4.3. Implementation Details

We train three stages separately and connect them to form a complete pipeline during inference. We construct transformers [49] based on linear transformer [26] to reduce time consumption. All three stages are trained with cross-entropy loss and teacher-forcing strategy. During inference, we use a stochastic temperature-controlled sampling [19] to increase the diversity of generated samples. We train Chord, Melody, and Accompaniment Transformers for 200, 200, and 400 epochs, respectively, on one V100 GPU. We use fluidsynth[3] to synthesize our MIDI into audio. More implementation details are in the Appendix.

5. Evaluation Metric

In this chapter, we first extend the vision-language CLIP [36] to the video-music domain and propose a new evaluation metric named Video-Music CLIP Precision (VMCP) to measure the video-music correspondence.

5.1. Video-Music CLIP

To build the video-music CLIP model, we adopt the design choice in [46], the state-of-the-art video-music retrieval model. Specifically, we first split the input music and video into fixed-length segments and use CLIP [36] and music tagging model [51] to extract visual and audio features separately. Given the extracted features, we adopt a transformer encoder as the video encoder and music encoder to explore contextual relations and learn a joint multi-modal embedding space. The model is trained with the InfoNCE contrastive loss [4] to map positive video-music pairs closer while pushing negative pairs further in the CLIP-based joint embedding space. Loss of videos v to music pieces m is defined as:

$$\mathcal{L}^{v\to m} = -\sum_{i}^{N}\sum_{l}^{L}\left[\log\frac{\exp\left[s(v_{i,l}, m_{i,l})/\tau\right]}{\sum_{j}^{N}\sum_{k}^{L}\exp\left[s(v_{i,l}, m_{j,k})/\tau\right]}\right],$$

where N denotes number of video-music pieces, L denotes number of segments, $s(\cdot)$ denotes cosine similarity, and τ is a learnable temperature parameter. The music-to-video loss $\mathcal{L}^{m\to v}$ is defined symmetrically.

[3] https://github.com/FluidSynth/fluidsynth

Methods	Video-Music Correspondence				Music Quality				
	P@5	P@10	P@20	AR	SC	PE	PCE	EBR	IOI
Real (SymMV)	-	-	-	-	0.986	4.197	2.633	0.023	0.184
CMT [9]	8.9	17.7	31.0	33.4	0.990	3.920	2.444	0.074	0.246
w/o semantic	11.6	23.9	42.0	26.1	0.955	2.892	2.310	**0.019**	0.358
w/o color	15.6	**26.6**	**44.8**	**25.1**	0.956	2.732	2.200	0.011	0.330
w/o motion	12.2	22.2	37.9	26.3	0.975	3.010	2.283	0.004	0.261
Video2music	10.8	19.7	33.3	30.0	0.981	**3.990**	**2.639**	0.010	0.229
Video2chord2music	13.7	23.1	43.6	26.0	0.996	2.497	2.036	0.081	0.985
V-MusProd	**15.7**	24.6	**44.8**	25.4	**0.983**	3.940	2.607	0.004	**0.174**

Table 2: **Objective evaluation on SymMV test set.** We evaluate video-music correspondence and music quality with VMCP and music quality metrics. P indicates Precision, where higher is better. AR indicates average rank, where lower is better. For music quality metrics, **closer** to Real is better.

To train this model, we need to collect plenty of video-music pairs and ensure that the training dataset should roughly cover the distribution of our dataset. Therefore, we download video clips from YouTube8M dataset [1] annotated as "music video" and obtain ∼20k video-music pairs. After training on the YouTube music video dataset, we fine-tune the model with a small learning rate on audio converted from SymMV to improve its retrieval performance further.

5.2. Video-Music CLIP Precision (VMCP)

Equipped with the pretrained video-music CLIP model, we design a retrieval-based metric similar to [52]. Given a generated music piece in MIDI format, we first synthesize it into audio and calculate the top-K retrieval accuracy from a pool of N candidate videos using the CLIP model. Specifically, we rank the cosine similarity between the generated sample $\hat{m}$ and its condition video v and $M-1$ random sampled videos v_i. We consider a successful retrieval if the ground truth video is ranked in the top-K place. We test the model using all generated samples and compute the success retrieval rate as the final precision score. We set $M = 70, K = 5, 10, 20$. We also calculate the average rank of the ground truth video, where a lower rank implies better correspondence. Overall, the proposed metric is able to measure how well the generated music aligns with the input video. We validate that it shows a high correlation with human judgments in the experiments.

6. Experiments

We conduct comprehensive experiments on our V-MusProd model. In Sec. 6.1, we introduce the compared method CMT [9]. In Sec. 6.2, we evaluate video-music correspondence with VMCP and music quality with objective metrics. In Sec. 6.3, we conduct a thorough subjective evaluation for video-music correspondence and music quality by user study. In Sec. 6.4, we ablate our design choices and highlight the importance of different features used in video controller to validate the effectiveness of proposed method. In Sec. 6.5, we train V-MusProd in unconditional setting and evaluate its music quality against previous symbolic music generation methods.

6.1. Compared Method

We compare V-MusProd with the state-of-the-art video background music generation method CMT [9], the first and only method to generate full-length background music for general videos. CMT uses purely rule-based video-music rhythmic relationships without paired video-music data. Other video-conditional music generation methods mostly focus on specific video types (*e.g.* dance videos) and require extra annotations (*e.g.* keypoints [56, 45]), which are unavailable in the general setting. We train CMT on SymMV to provide a benchmark.

6.2. Objective Evaluation

Metrics. For video-music correspondence, we use VMCP to evaluate objectively, where higher precision and lower average rank are better. For music quality, we select music objective metrics from [11, 53], including Scale Consistency (SC), Pitch Entropy (PE), Pitch Class Entropy (PCE), Empty Beat Rate (EBR), and average Inter-Onset Interval (IOI), which evaluate music by pitches and rhythm. Note that these music quality metrics are not indicated by how high or low they are but instead by their *closeness* to the real data. We use SymMV test set for evaluation.

Results. As shown in Tab. 2, V-MusProd surpasses CMT on VMCP and music quality metrics. This proves our method achieves better video-music correspondence and music quality than the state-of-the-art method.

Metrics	Expert	Non-expert
Music Melody	77%	82%
Music Rhythm	63%	53%
Video Content	63%	63%
Video Rhythm	60%	57%
Chord Quality	63%	-
Accom. Quality	83%	-
Overall Ranking	73%	67%

Table 3: **Subjective evaluation for V-MusProd against CMT [9].** We show preference rates in music quality metrics, video-music correspondence metrics, and expertise metrics.

6.3. Subjective Evaluation

Subjective evaluation is widely adopted in previous works [9, 21, 54, 22]. We conduct a user study by sending out questionnaires. We invite 55 participants, including 10 *experts* with expert knowledge in music composition and 45 *non-experts*. We provide several videos from different categories like scenery, city scenes, and movies. Each video has two pieces of background music generated by V-MusProd and CMT, presented randomly for blindness. The questionnaire takes about 20 minutes to complete.

Metrics. Participants are required to compare two background music pieces from several aspects: (1) Music Melody: melodiousness and richness of music theme; (2) Music Rhythm: structure consistency of rhythm; (3) Video Content: correspondence between video content and music; (4) Video Rhythm: correspondence between video motion and music rhythm; (5) Overall Ranking: overall preference of the two samples. Besides, we ask the expert group to evaluate two additional metrics related to music theory: (6) Chord Quality: the quality of chord progression in the music; (7) Accompaniment Quality: the richness of music accompaniment.

Results. We provide results of preference rate, *i.e.* the percentage of users who consider our music better than CMT, in Tab. 3. Results show that V-MusProd outperforms CMT ($> 50\%$) in nearly all metrics and user groups, demonstrating better music quality and correspondence with videos. In particular, our model outperforms CMT in both Music Melody and Accompaniment Quality by a large margin, indicating our decoupling generation of melody and accompaniment significantly improves their qualities. The subjective evaluation results show high correlations with VMCP, which further verifies the effectiveness of our proposed metric.

6.4. Ablation Study

Ablation on Video Controller. We ablate the three video control features and evaluate the video-music correspondence with VMCP: (1) w/o semantic: no semantic feature input for video controller; (2) w/o color: no color feature input for video controller; (3) w/o motion: fix tempo and do not add timing encoding. As shown in Tab. 2, semantic and motion features are significant for correspondence. We observe that w/o color has similar correspondence with the full model despite lower music quality. We attribute this to the fact that music tonality is connected with the color of videos. The original keys have already recorded the information on video colors, so color features are unnecessary for video-music correspondence modeling. If we remove remove the influence of tonality by changing keys, we need color features to capture the video colors.

Ablation on Music Generator. We further conduct ablation study on our music generator with VMCP. To verify the necessity of the decoupled structure, we test two variants of our model: (1) Video2music: uses the output video features of the fusion encoder to directly generate target music by a Transformer decoder without decoupling the structure of chords, melody, and accompaniment; (2) Video2chord2music: generate chords first and then use chords to generate music without decoupling melody and accompaniment. As shown in Tab. 2, removing any one or more components of chords, melody, and accompaniment hurts the overall performance of correspondence while having similar music quality. The difference in correspondence and music quality validates that decoupled structure is important for music generation and imposing video controls. The improvement of VMCP from Video2music to Video2chord2music shows the effectiveness of decoupling chords, and the improvement from Video2chord2music to the full model V-MusProd shows the effectiveness of decoupling melody and accompaniment. We note that Video2music sometimes has better music quality. It can be explained that imposing control over music generation can hurt music quality by adding inductive biases.

Methods	SC	PE	PCE	EBR	IOI
Real (POP909)	0.965	4.455	2.774	0.005	0.125
CP [21]	0.987	3.697	2.538	0.041	0.250
Music Trans. [22]	0.985	3.934	2.581	0.034	0.216
HAT [54]	0.989	3.856	2.550	0.040	**0.139**
V-MusProd	**0.967**	**4.070**	**2.774**	**0.005**	0.171

Table 4: **Results of unconditional generation on POP909 [50].** For all the metrics, **closer** to Real is better.

6.5. Unconditional Generation

Our method can be directly used in unconditional music generation. We examine V-MusProd against previous music generation methods: (a) HAT [54]: a hierarchical model built on multiple transformer-based levels to enhance the structure of music, achieving state-of-the-art generation quality; (b) CP Transformer [21]: transformer-based model using 2D

music tokens to compress sequence length; (c) Music Transformer [22]: the first transformer-based music generation model with improved relative attention.

All the above methods are trained on POP909[50] dataset. We directly use their publicly available demos for evaluation. We train our V-MusProd on POP909 without video input, *i.e.* training Chord Transformer without cross attention with video features. The unconditionally generated results are evaluated by music quality metrics in Sec. 6.2. As shown in Tab. 4, our V-MusProd achieves closer results to POP909 training set than previous methods for most of the metrics. This indicates that unconditional music generation can benefit from our decoupling structure.

7. Conclusion

In this paper, we have introduced the SymMV dataset, which contains 1140 videos and corresponding background music with rich annotations. Based on SymMV, we developed a benchmark model V-MusProd. It decouples music into chords, melody, and accompaniment, then utilizes video-music relations of semantic, color, and motion features to guide the generation process. We also introduced the VMCP metric based on video-music CLIP to evaluate video-music correspondence. With VMCP and subjective evaluation, we prove that V-MusProd outperforms baseline model CMT in correspondence both qualitatively and quantitatively.

Acknowledgement This work was supported in part by the National Key R&D Program of China under Grant 2022ZD0115502, the National Natural Science Foundation of China under Grant 62122010, and the CCF-DiDi GAIA Collaborative Research Funds for Young Scholars.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. Histogan: Controlling colors of gan-generated and real images via color histograms. In *CVPR*, 2021.

[3] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[4] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-Supervised MultiModal Versatile Networks. In *NeurIPS*, 2020.

[5] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new Python Audio and Music Signal Processing Library. In *MM*, 2016.

[6] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

[7] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020.

[8] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[9] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *MM*, 2021.

[10] Tan M Dinh, Rang Nguyen, and Binh-Son Hua. Tise: Bag of metrics for text-to-image synthesis evaluation. In *ECCV*, 2022.

[11] Hao-Wen Dong, Ke Chen, Julian McAuley, and Taylor Berg-Kirkpatrick. Muspy: A toolkit for symbolic music generation. In *ISMIR*, 2020.

[12] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *AAAI*, 2018.

[13] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021.

[14] Chuang Gan, Deng Huang, Peihao Chen, and Joshua B Tenenbaum. Foley music: Learning to generate music from videos. In *ECCV*, 2020.

[15] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[16] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

[17] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the maestro dataset. In *ICLR*, 2019.

[18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[19] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020.

[20] Sungeun Hong, Woobin Im, and Hyun S Yang. Content-based video-music retrieval using soft intra-modal structure constraint. *arXiv preprint arXiv:1704.06761*, 2017.

[21] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *AAAI*, 2021.

[22] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *ICLR*, 2019.

[23] Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023.

[24] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *MM*, 2020.

[25] Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

[26] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[27] A Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *ICASSP*, 2020.

[28] Stefan Kostka and Dorothy Payne. *Tonal harmony*. McGraw-Hill Higher Education, 2013.

[29] Carol L Krumhansl. *Cognitive foundations of musical pitch*. Oxford University Press, 2001.

[30] Sangho Lee, Jiwan Chung, Youngjae Yu, Gunhee Kim, Thomas Breuel, Gal Chechik, and Yale Song. Acav100m: Automatic curation of large-scale datasets for audio-visual video representation learning. In *ICCV*, 2021.

[31] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *TMM*, 2018.

[32] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021.

[33] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019.

[34] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[35] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *NeurIPS Datasets and Benchmarks Track*, 2021.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[37] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[38] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation. In *MM*, 2020.

[39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[40] Arnold Schoenberg. *Theory of harmony*. University of California Press, 1983.

[41] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[42] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *SIGIR*, 2021.

[43] Kun Su, Xiulong Liu, and Eli Shlizerman. Audeo: Audio generation for a silent performance video. In *NeurIPS*, 2020.

[44] Kun Su, Xiulong Liu, and Eli Shlizerman. Multi-instrumentalist net: Unsupervised generation of music from body movements. *arXiv preprint arXiv:2012.03478*, 2020.

[45] Kun Su, Xiulong Liu, and Eli Shlizerman. How does it sound? In *NeurIPS*, 2021.

[46] Dídac Surís, Carl Vondrick, Bryan Russell, and Justin Salamon. It's time for artistic correspondence in music and video. *CVPR*, 2022.

[47] Alexandra Uitdenbogerd and Justin Zobel. Melodic matching techniques for large music databases. In *MM*, 1999.

[48] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[50] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia. Pop909: A pop-song dataset for music arrangement generation. In *ISMIR*, 2020.

[51] Minz Won, Keunwoo Choi, and Xavier Serra. Semi-supervised music tagging transformer. In *ISMIR*, 2021.

[52] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[53] Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. *Neural Computing and Applications*, 2020.

[54] Xueyao Zhang, Jinchao Zhang, Yao Qiu, Li Wang, and Jie Zhou. Structure-enhanced pop music generation via harmony-aware learning. *arXiv preprint arXiv:2109.06441*, 2021.

[55] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

[56] Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. Quantized gan for complex music generation from dance videos. In *ECCV*, 2022.

[57] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal and conditional generation. *arXiv preprint arXiv:2206.07771*, 2022.